

MySQL Failovers using Python

```
./mysql_failover_db.py -d -e app_read -t west_db#
```

script: `./mysql_failover_db.py`

default options: `-d`

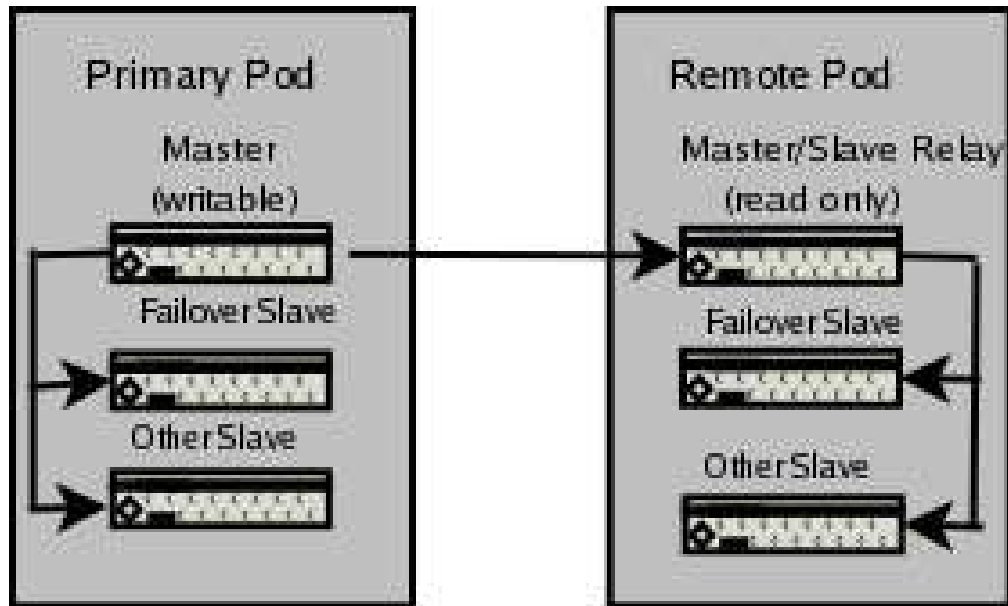
execute command: `-e <command>`

target servers: `-t <servers>`

source servers: `-s <servers>`

= All servers of this type.

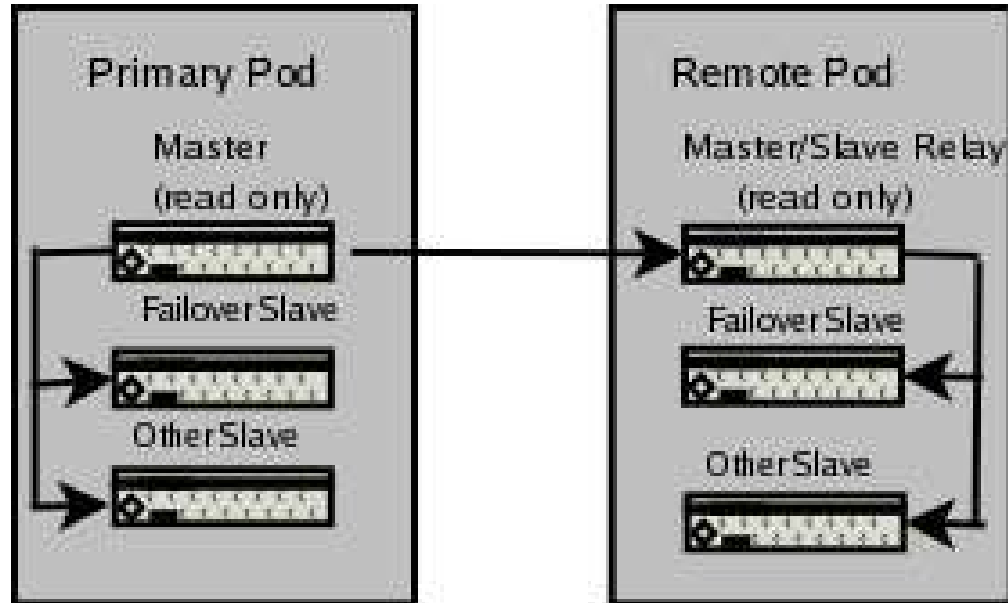
MySQL Failovers using Python



Goals:

1. Perform Failover within 1 hour with downtime.
2. Automate commands with Python script.
3. Perform FailBack later with 1 hour of downtime.
4. Perform on multiple MySQL Master servers.

MySQL Failovers using Python

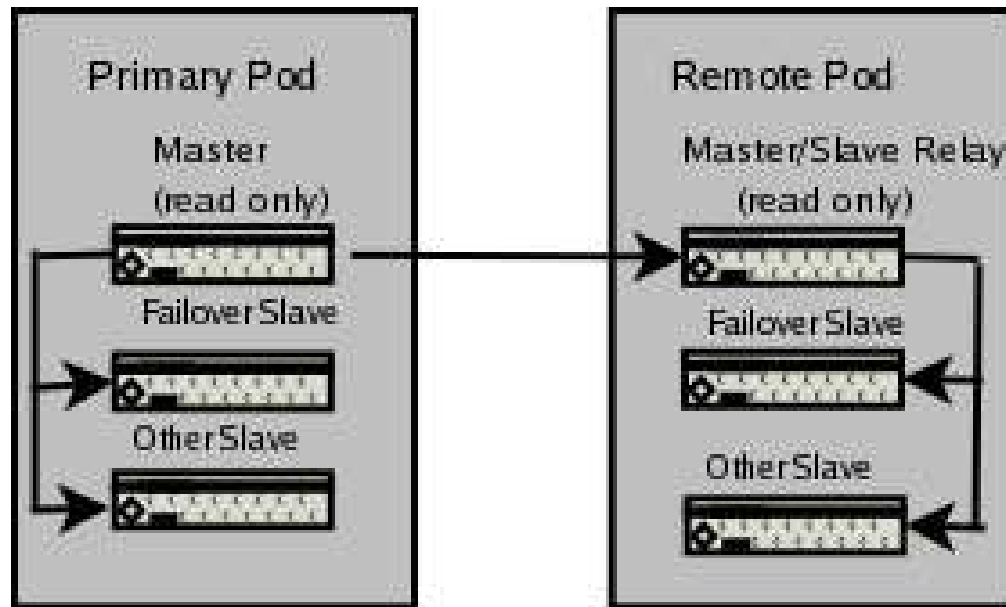


1. Stop all Applications
2. Change Permissions on Primary Master to read only.



Command executed: `./mysql_failover_db.py -d -e app_read -t west_db#`

MySQL Failovers using Python

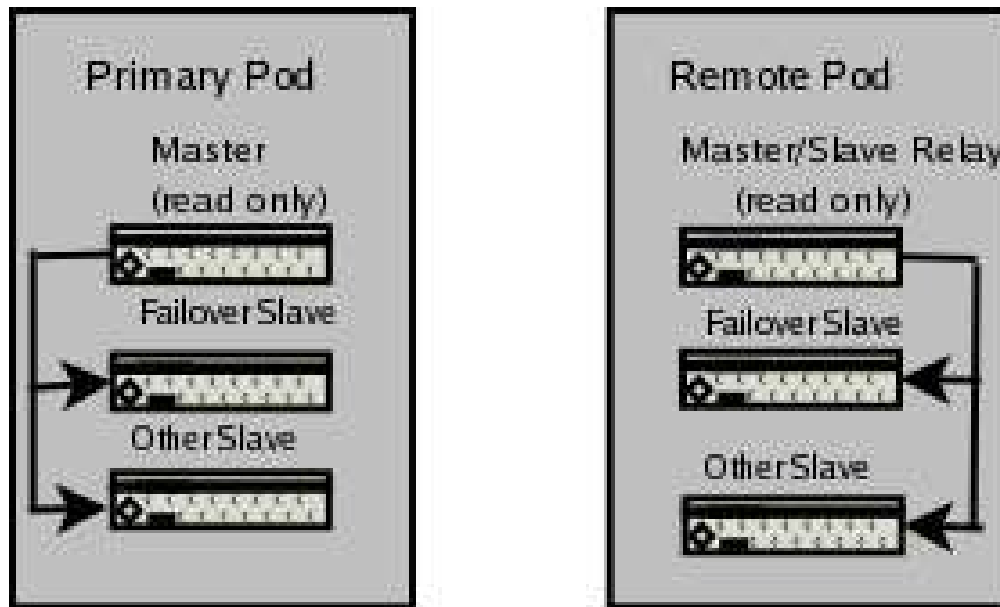


1. Check replication status.



Command executed: `./mysql_failover_db.py -d -e slave_status -a`

MySQL Failovers using Python



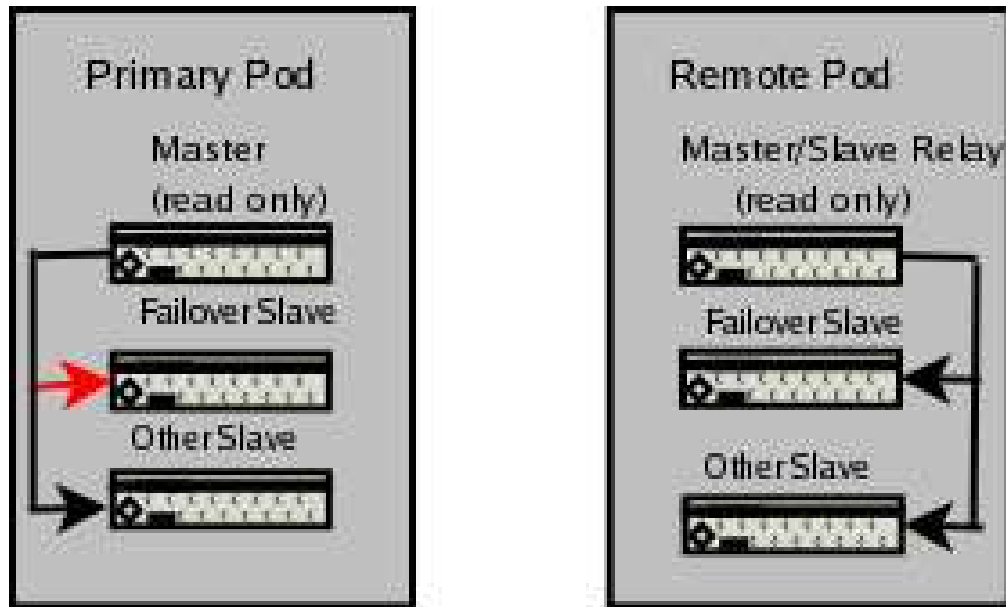
1. Stop Replication Remote Pod
2. Delete Replication to Remote Pod



Command executed: `./mysql_failover_db.py -d -e stop_replication -t east_db#`

Command executed: `./mysql_failover_db.py -d -e delete_replication -t east_db#`

MySQL Failovers using Python

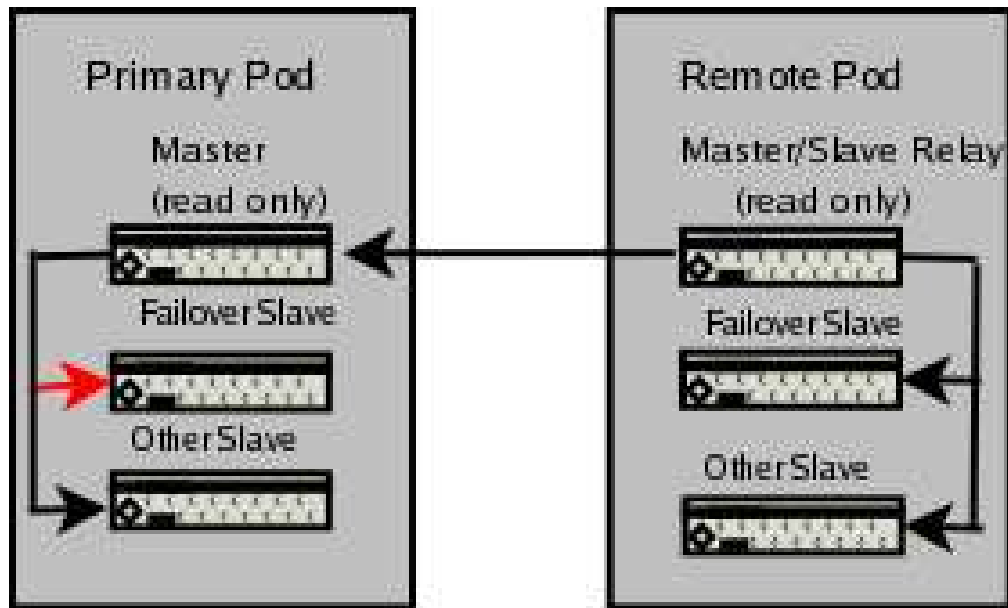


1. Perform Backups of Primary Pod databases.
2. Turn off Replication to Failover Slave in Primary Pod



Command executed: `./mysql_failover_db.py -d -e stop_replication -t west_db_failover#`

MySQL Failovers using Python



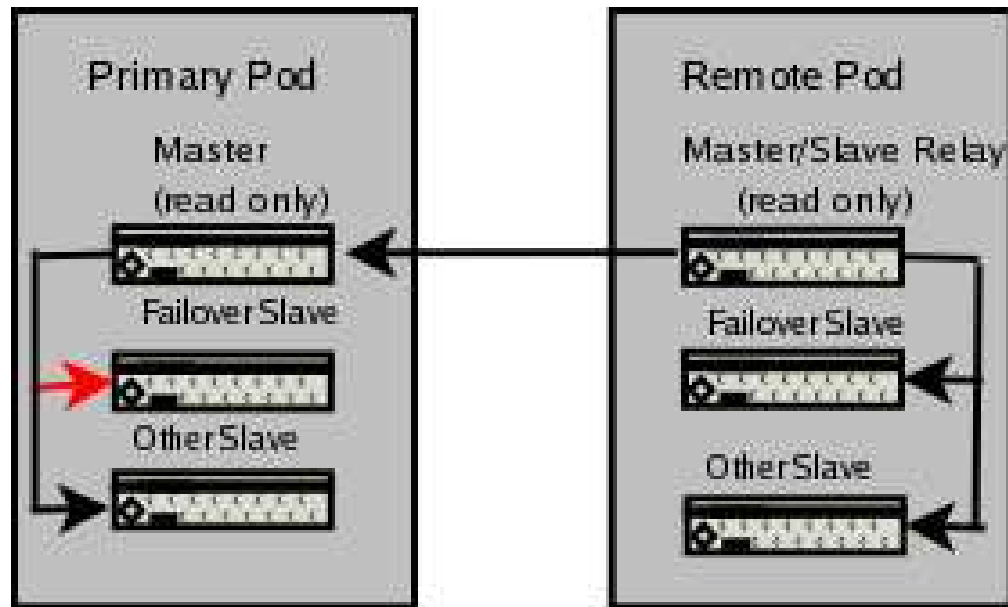
1. Replicate from the east coast to the west coast or, Remote Pod to the Primary Pod.



```
./mysql_failover_db.py -d -e change_master -t west_db# -s east_db#
```

```
./mysql_failover_db.py -d -e start_replication -t west_db#
```

MySQL Failovers using Python

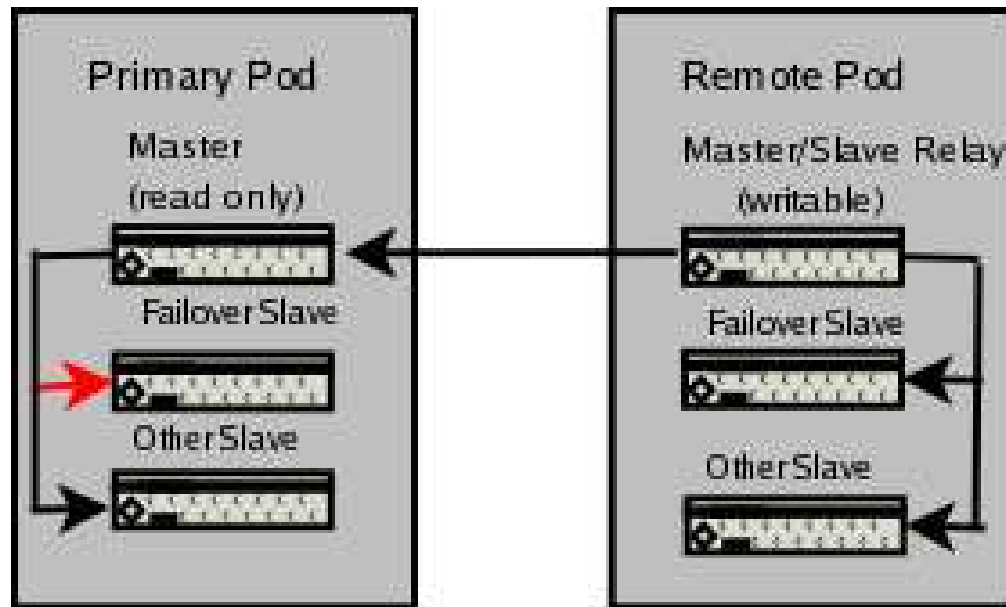


1. Check replication status.



```
./mysql_failover_db.py -d -e slave_status -a
```

MySQL Failovers using Python

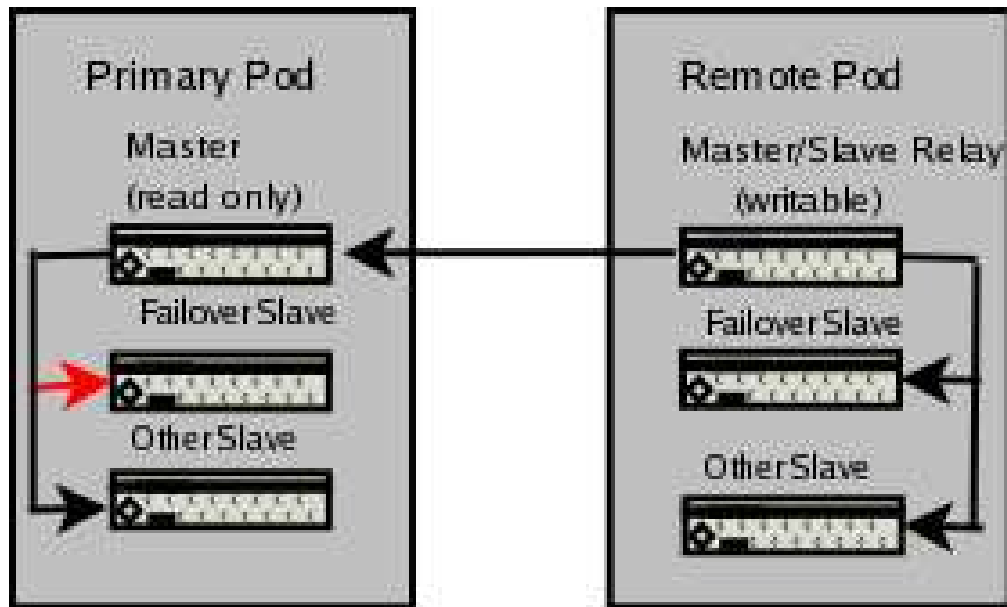


1. Change permissions on "application" account in MySQL to writable in the Remote or west coast Pod.



```
./mysql_failover_db.py -d -e app_write -t east_db#
```

MySQL Failovers using Python

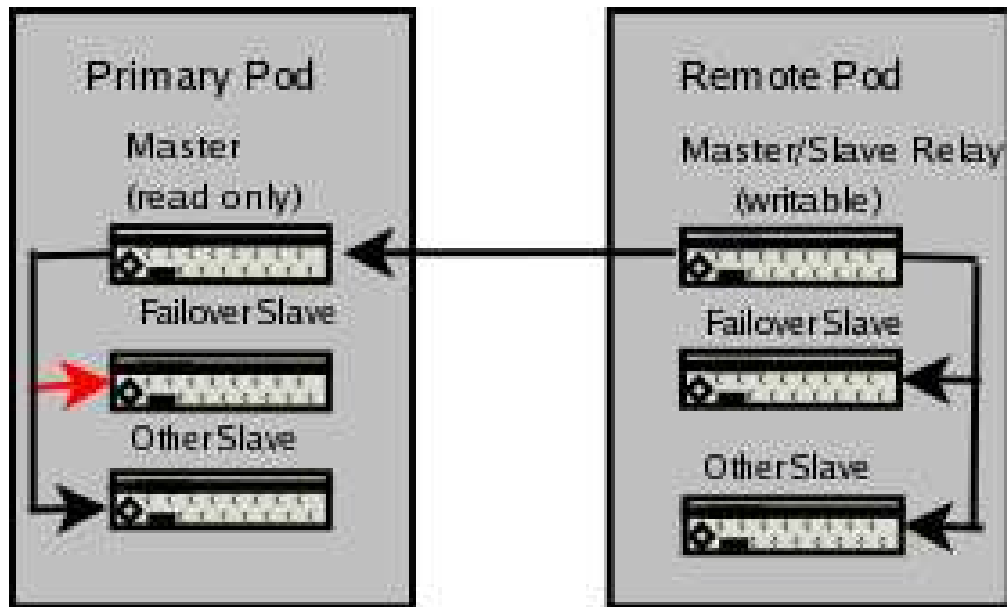


1. Test Replication using a noop table.



```
./mysql_failover_db.py -d -e check_replication -t west_db# -s east_db#
```

MySQL Failovers using Python



1. Any DNS entries would have been changed by now for the applications. No DNS entries were changed for the databases.
2. Start applications and test.

